

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Previously presented) A method for facilitating typesafe software
2 design while supporting structured composition of a software system, comprising:
3 receiving a first invocation of the software system;
4 assigning a first context to the first invocation;
5 examining the first invocation to locate components of the first invocation;
6 registering a unique factory to build each component in a software design
7 environment, wherein the unique factory is registered using the first context;
8 providing an additional factory for building an extended component of the
9 first invocation; and
10 when a component is needed, building the component using the unique
11 factory associated with the component, whereby building the component after
12 each component has a registered factory eliminates potential problems with
13 initialization circularity.

- 1 2. (Previously presented) The method of claim 1, further comprising:
2 receiving a second invocation of the software system;
3 assigning a second context to the second invocation;
4 examining the second invocation to locate components of the second
5 invocation;

6 registering a unique factory to build each component in the software
7 design environment, wherein the unique factory is registered using the second
8 context; and
9 when a component is needed, building the component using a factory
10 associated with the component, whereby building the component after each
11 component has a registered factory eliminates problems with initialization
12 circularity.

1 3. (Original) The method of claim 2, wherein components from the second
2 invocation are not available to the first invocation.

1 4 (Canceled).

1 5. (Previously presented) The method of claim 1, wherein registering the
2 unique factory to build each component in the software design environment
3 involves placing a key and a related factory identifier into a storage structure.

1 6. (Previously presented) The method of claim 5, wherein building the
2 component in the software design environment using the unique factory
3 associated with the component involves using the key to retrieve the related
4 factory identifier from the storage structure.

1 7. (Original) The method of claim 6, wherein the storage structure
2 includes a hash table.

1 8. (Previously presented) A computer-readable storage device storing
2 instructions that when executed by a computer cause the computer to perform a

3 method for facilitating typesafe software design while supporting structured
4 composition of a software system, the method comprising:
5 receiving a first invocation of the software system;
6 assigning a first context to the first invocation;
7 examining the first invocation to locate components of the first invocation;
8 registering a unique factory to build each component in a software design
9 environment, wherein the unique factory is registered using the first context;
10 providing an additional factory for building an extended component of the
11 first invocation; and
12 when a component is needed, building the component using the unique
13 factory associated with the component, whereby building the component after
14 each component has a registered factory eliminates potential problems with
15 initialization circularity.

1 9. (Previously presented) The computer-readable storage device of claim
2 8, the method further comprising:
3 receiving a second invocation of the software system;
4 assigning a second context to the second invocation;
5 examining the second invocation to locate components of the second
6 invocation;
7 registering a unique factory to build each component in the software
8 design environment, wherein the unique factory is registered using the second
9 context; and
10 when a component is needed, building the component using a factory
11 associated with the component, whereby building the component after each
12 component has a registered factory eliminates problems with initialization
13 circularity.

1 10. (Previously presented) The computer-readable storage device of claim
2 9, wherein components from the second invocation are not available to the first
3 invocation.

1 11 (Canceled).

1 12. (Previously presented) The computer-readable storage device of claim
2 8, wherein registering the unique factory to build each component in the software
3 design environment involves placing a key and a related factory identifier into a
4 storage structure.

1 13. (Previously presented) The computer-readable storage device of claim
2 12, wherein building the component using the unique factory associated with the
3 component involves using the key to retrieve the related factory identifier from
4 the storage structure.

1 14. (Previously presented) The computer-readable storage device of claim
2 13, wherein the storage structure includes a hash table.

1 15-21 (Canceled).